

②



1698

92

Approved for public release  
Distribution Unlimited

## **VLSI for High-Speed Digital Signal Processing**

### **Quarterly Progress Report – 7/1/92 through 9/30/92**

During the present quarter we have tested our multiplier test chip fabricated by MOSIS. This IC is a TinyChip comprising the 12-bit by 11-bit multiplier, the coefficient and data input registers, the output register, and RAM to store the coefficient and input data. Figure 1 shows the architecture for the multiplier test IC. The multiplier itself consists of 3100 transistors occupying an area of  $1.53 \text{ mm}^2$  (1.313 mm by 1.166 mm) in  $2\text{-}\mu\text{m}$  CMOS technology and is simulated to operate in 22 ns. The chips were tested using a Tektronix LV-500 Logic Verification System. All 4 TinyChips were fully functional and showed a worst-case delay of 22.5 ns, agreeing well with the simulated results. We have also received from MOSIS our register block test chip and are in the process of testing it. This test IC is also a TinyChip and contains a complete 16-bit by 16-word dual-port register block as well as input and output registers to allow the critical path to be measured accurately. The register block was redesigned during the previous quarter in order to achieve a 50 MHz operating speed. It consists of 4000 transistors and occupies an area of  $1.61 \text{ mm}^2$  (1.284 mm by 1.256 mm) in  $2\text{-}\mu\text{m}$  CMOS technology. We expect the test results will match the simulated performance. Thus, the overall speed for the entire five-processor system will only be constrained by the cycle time of the multiplier and should exceed 40 MHz.

We are also currently completing the design and layout of the remaining portions of

the ALU. Simulation results to date indicate that this circuitry will operate in excess of 50 MHz and therefore will not impact the overall performance of the system. We expect to complete this design and to submit to MOSIS a chip containing a complete processor and two register blocks at the end of the next quarter.

A paper detailing the specific implementation issues involved in the design of the multiplier has been submitted for publication in the *IEEE Journal of Solid State Circuits*. In addition, we have presented a paper on the five-processor system at the International Symposium for Systems, Signals, and Electronics in Paris, France from September 1-4, 1992. We will also be presenting a paper on the VLSI implementation of this system at the 1992 IEEE Workshop on VLSI Signal Processing in Napa, California on October 28-30, 1992.

In addition to the issues just discussed, which concern the chip's hardware, our project also has a software component. Here we are running ahead of schedule. Sept 9 (mentioned in the Year 2 statement of work) concerns the development and refinement of the task partitioner, which is the program that analyzes an arbitrary digital filter structure and automatically writes programs for our ring of parallel processors. We have now completed writing and testing the first version of this software. It performs correctly, in that it is capable of correctly writing programs for all filter structures that we have tested it on. We intend to continue to make improvements to this software during the ensuing course of this project. We have just recently found ways of improving its speed of operation, which was initially a bit slow (20 to 25 minutes, on an IBM RT) when writing optimal parallel-processing programs for some complex

filter structures. Our improvements have now yielded operating speeds of at most several minutes for all programming tasks attempted. Additional features will be added to the program, and its user interface will be improved during Year 2.

Furthermore, we have submitted a paper for publication concerning our programming algorithm. It explains that the programs that it computes are optimal; that is they use the minimum number of program steps per data sample to implement a given arbitrary digital filter structure.

The algorithm's "input" is a SPICE-like description of the desired digital filter structure. The algorithm first combines each multiplication with a subsequent addition into a two-step Multiply-Accumulate instruction. Using a well-known technique of Renfors and Neuvo, it calculates the theoretical minimum sampling period that would be possible for any custom parallel implementation of the specified filter structure assuming that an unlimited number of processors are available. It then calculates the optimum sampling period for the implementation of the filter structure on a multiple processor system with  $P$  processors. Our algorithm detects all multiple-input adders in the desired filter structure and provides the user the option of searching for the optimum adder sequence to minimize the filter's sampling period.

Having found the optimum sampling period the next task is to determine the time schedule. The earliest time at which an operation can be started is found from a maximal distance spanning tree, which is found using an algorithm similar to the Bellman-Ford method. Since all operations are executed periodically we modify the time schedule by specifying its values modulo the sampling period.

We next determine how the operations will be distributed over the processors. If we find that it is not possible to appropriately distribute the operations over the processors, the operations will be rescheduled; that is, one of the operations will be selected to start at a different time (a different step in the program).

After the operations are distributed over the processors the computer algorithm checks whether all data can be made accessible to all processors that need it. If this check fails a redistribution assigns an operation to the next available processor. In this way all possible distributions of the operations over the parallel processors in the ring-structured DSP chip can be tested. The "output" of the computer algorithm is a set of programs for the parallel processors which causes them to implement the given filter structure.

To speed up the task partitioner's distribution of operations over the processors and to reduce the communication between processors our algorithm has the option to add the additional constraint that an operation may only be assigned to that processor where its input data are created, or one of the two adjacent processors.

While we have been able to implement all practical examples of digital filter structures at the optimum sampling period, it is possible to create contrived data-flow graphs for which an implementation on  $P$  processors at the optimum sampling period does not exist. If that occurs the sampling period will be increased by 1 time unit, and the algorithm will then continue with the initial time scheduling.

Three examples are given in the paper to illustrate the working of the algorithm, and a comparison with other scheduling algorithms is made.

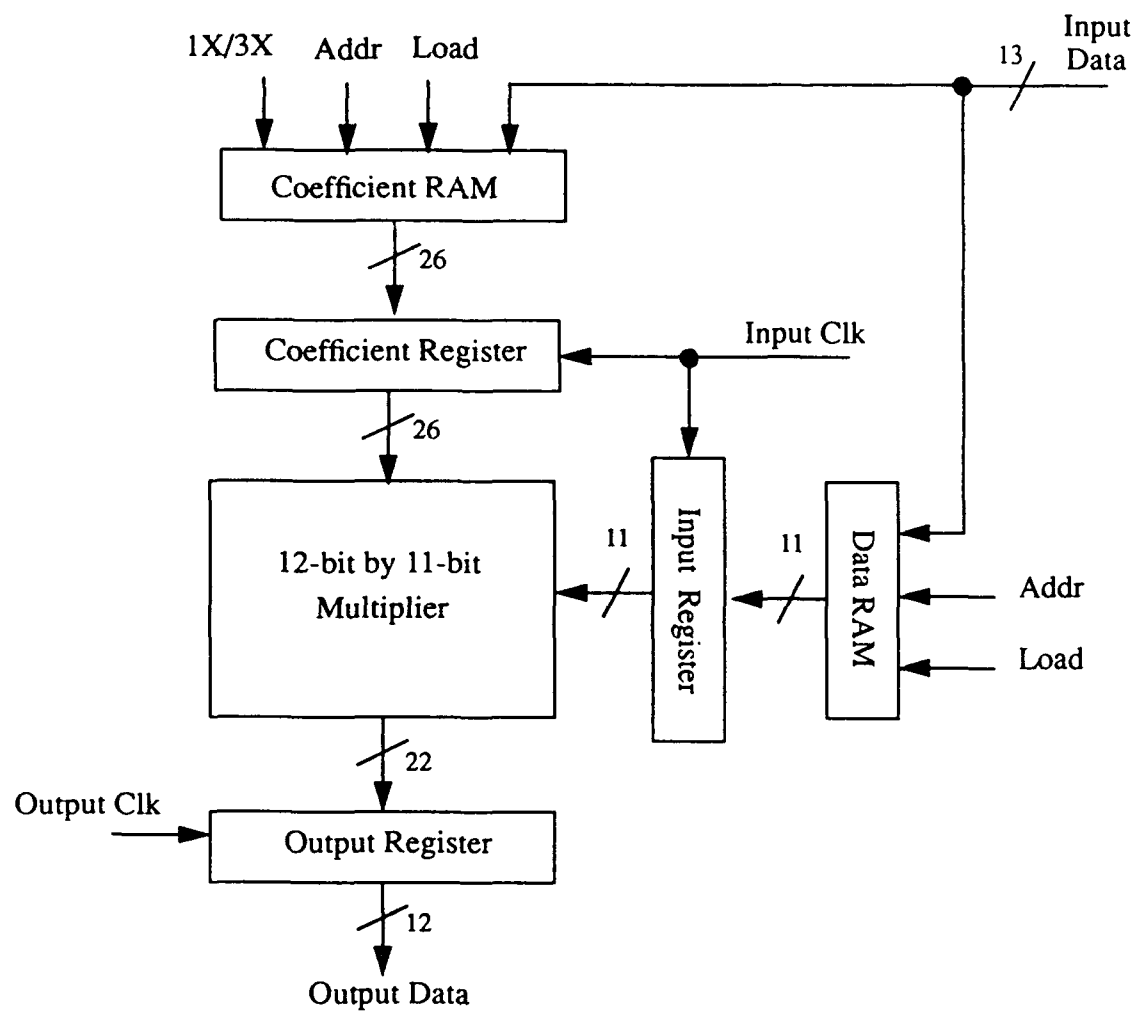


Figure 1 - Block Diagram of Multiplier Test IC